Wireless Pers Commun (2017) 96:5185–5201
DOI 10.1007/s11277-016-3735-5

An Augmented Routing Algorithm for Trusted Detection of Link Failures in MANETs

M. S. Rahul[1] · E. Arun[1] · P. Mohamed Shameem[1] · J. Rajeesh[2]

Published online: 5 October 2016

Abstract Now a days, the communication between different nodes in a Mobile Ad hoc Network (MANET) is not guarded. Various encryption mechanisms are used to protect the communication between nodes. Link failures and packet dropping due to unfaithful nodes are becoming one of the main opposition for the trusted detection of malicious nodes. A failure can occur either due to channel errors or harmful nodes in network. These attacks may have the intention of modifying the routing protocol so that the data transmission through a specific node controlled by the attacker disturbs the network topology. Thus it deteriorates the performance of network. Mutual association of dropped packets is capitalized for synthesizing the suspicious nodes in MANET. The algorithm proposed is using an efficient cryptosystem with cipher text list validator scheme and a communal auditing scheme for the validation of certificate received from individual nodes. For constructing the framework, the proposed algorithm with five phases has a network setup phase, data routing phase, communal auditing phase, error node detection phase and a data receiver phase. This framework makes the MANET node build a safe routing topology by effectively judging the harmful nodes as well as the unfaithful information accepted from supplementary nodes.

Keywords MANET · Packet dropping · Secure routing · Cipher text list validator (CLV) · Communal auditing

✉ M. S. Rahul
rahulms23@gmail.com

✉ E. Arun
drearun@yahoo.com

[1] Department of Computer Science and Engineering, T.K.M Institute of Technology, Kollam, Kerala, India

[2] Department of Electronics and Biomedical Engineering, T.K.M Institute of Technology, Kollam, Kerala, India

1 Introduction

A wireless multi hop network is an infrastructure less network and is unprotected due to intrinsic attributes of such networks. Every node in a MANET is free to move independently. Each must forward traffic unrelated to its own use, and therefore be a router. In MANETs each device need to maintain information continuously due to the mobility of nodes. Due to the high mobility packet loss during transmission is becoming vulnerable. In order to avoid data loss, a secure routing scheme should be a key factor for a MANET. Packet-dropping can be occurred due to several reasons such as due to the presence of a malicious node, due to the presence of unhealthy channel conditions (e.g., fading, noise, and interference). To solve all these issues, an anonymous secure routing [1] is needed. Continuous dropping of packets causes performance degradation of the whole network. But these kinds of attacks are easy to be identified [23] due to the continuous presence of malicious drop. In some cases the presence of attack can be found out but the actual node which causes the attack is not recognized. In such cases, a cooperative node can use random dispersive route [22] to overcome the fake reply from a malicious node.

Different kinds of group signature schemes [9] are introduced for a scalable packet delivery. A digital group signature scheme can perform authentication without revealing node identities. Common methods widely used for secure routing are trapdoor [16], onion routing [8] and group signature. We now focus on attacks that affect the routing protocol in ad hoc networks. Such attacks are having the intention of changing the routing protocol for controlling the data flow through a specific node. An attack may also have a plan to break the formation of the network, making genuine nodes store fake routes, and more generally disturbs the network topology. Routing level attacks can be classified into two main classes: faithful traffic generation and unfaithful traffic forwarding. In some cases, these two classes coincide with misbehaving nodes that are not due to maliciousness, e.g. node failure, battery depletion, or radio interference. The first category includes attacks which consist of sending false control messages destined for another node, or control messages which contain inaccurate routing information. The network may exhibit Byzantine behavior [10, 11], i.e. conflicting information in different sections of the network. The outcome of this attack forms degradation in network communications, unreachable nodes, and possible routing loops.

The second category includes Black hole attack, message tampering, replay attack, wormhole attack and rushing attack. Network data transmission coming from legitimate protocol nodes may be infected by misbehaving nodes. In *Black hole attack*, an intruder can leak received routing messages, rather than forwarding them as the requirement of protocol. This reduces the contents of routing information available to the other nodes. These attacks are passive in nature and a simple way to perform a Denial of Service (DOS). The attack can perform selectively such that they drop routing packets for a specified destination, or a randomly selected portion of the packet or drop all packets. This makes the destination node unreachable or degrades communication through the network. In *message tampering*, attacker manipulates the messages originating from other nodes before transferring them and it does not be the digest of the payload. In replay attack, as topology changes, old control message describes topology configuration were no longer exists, even though if it is valid in the past. An attacker can perform a replay attack by recording old valid control messages and retransmit them to make other nodes change their routing tables with old routes. This attack is successful without a timestamp even if control messages exhibit a digest or a digital signature.

The *wormhole attack* is quite vulnerable and consists of traffic monitoring from one area of the network and replaying it in a different area. The severity of the wormhole attack shows different properties such that it is difficult to detect and is effective even in a network where privacy, integrity, authentication, and non-repudiation are preserved. Furthermore, wormholes are very likely to be chosen as routes on a distance vector routing protocol because they provide a shorter path to the destination. A disgrace that can be carried out against on-demand routing protocols is the *rushing attack*. In general, on-demand routing protocols defines that nodes only forward the first received route request from each route discovery process and all further received route requests are ignored. The work in [17] describes an efficient communication protocol over P2P applications. It achieves better computational and energy consumption during route discovery by the probabilistic flooding of route request packet without using a hop-by-hop encryption. When a route discovery is initiated, an adversarial attack quickly forwards route request messages. If the first route requests reach the destination's neighbors from the attacker, then any discovered route that contains the attacker.

In this paper, we develop an augmented routing algorithm for the trusted detection of link failures on the basis of the accurate detection of malicious nodes. Our algorithm achieves high detection accuracy with the help of proof of acceptance database from individual nodes and the interrelationship between the positions of lost packets which is calculated from the auto-association function of the packet-loss bitmap. A packet-loss bitmap is used for indicating the status of sending packet. The status of packet, whether it is lost or received will be audited by using an independent auditing module. The auditor validates the bitmap and identifies the intruder and publishes the result to the requested node. Due to the independent auditing methodology, it helps to reduce the computation overhead during the operation. The main problem that we faced during the development of the algorithm lies in how to ensure that the packet-loss bitmaps reported by individual nodes are trusted or not. This reflects the current status of each packet, whether it is accepted or rejected. A cipher text list validator scheme is used to provide a proof of storage to the client nodes under adversarial conditions.

There have been many topology based and location based routing protocols are introduced for the past decade. But all of them are vulnerable in accurately detecting harmful nodes in MANET. The routing protocols like AODV [18], ANODR [19], DSR [20], ALERT [21] and AASR does not have the detection accuracy for finding a malicious node. It is very difficult to provide a trusted communication between nodes in a mobile ad hoc network. Most of the protocols are focusing on packet drop rate and the probability of dropping. Packet leaking inside a MANET can be occurred intentionally or unintentionally. On a wireless ad hoc medium, the chance of occurring packet loss is high due to noise, interference, and channel error or by some inside/outside attackers. For predicting the reason behind the packet dropping or the link failure under these circumstances will be more critical. So we need an efficient routing algorithm for transferring data securely under adversarial conditions.

Our framework provides privacy conservation without revealing the node identities. The packets are transferred across the route through the auditing information given by individual nodes. Due to the increase in individual reports submitted by nodes will not affect the privacy of auditing process. A cipher text list validator (CLV) signature [2, 7] based algorithm with a communal auditing scheme is developed to achieve low communication and storage overhead between source and destination. The proposed scheme allows one to achieve a balance in detection accuracy for malicious nodes and link failures.

The proposed detection scheme increases the scalability of cryptographic operations. Due to this nature, we can reduce the delay in cryptographic operations. While observing the proposed solution with respect to others, it shows a significant reduction in transmission delay along intermediate nodes. This scheme also allows routing protocols like AASR, AODV, and DSR for improving their detection accuracy on malicious nodes and link failures in a distinguished manner. In our scheme, communal auditor plays a crucial role which makes the auditing of information based on the report submitted by nodes in periodic time.

The remainder of this paper is organized as follows. In Sect. 2 we introduce the background and related work for the detection of harmful nodes. The network models, adversary models and problem statement are demonstrated in Sect. 3. We present the proposed scheme details with five phases of operation in Sect. 4. The effect of proposed algorithm for network security is analyzed in Sect. 5. Computation, communication and storage overheads occurred while using the proposed scheme are summarized in Sect. 6. Simulation setup and results are presented in Sect. 7, and we conclude the paper in Sect. 8.

2 Background and Related Work

Here we present some of the previous schemes or concepts used for secure routing in an adversarial environment. Depending on the accuracy of malicious node detection, the literature survey of this paper primarily focuses on four methods based on the different schemes used on it.

In the first method, a malicious node will receive good number of credits by sending most of the packets that it receives from upstream nodes. A credit system [3] provides an incentive for cooperation. A node gets credit by transferring packets for others, and uses its credit to send its own packets. As a result, a harmful node that continuous to drop packets will eventually remove its credit, and will not be able to send its own traffic. In the second method, the malicious node is capable of maintaining a very good reputation by forwarding most of the packets to its neighbor node. A reputation system [6, 13] depends on neighbors to monitor and identify misbehaving nodes. A node with high packet leaking rate will get a bad reputation by its neighbors. This information is transmitted periodically along the network and is preserved as an important parameter for selecting routes. CONFIDANT protocol proposed by Buchegger and Le Boudec in [12] is an example for a reputation based scheme.

In the third method, actual malicious attack which causes a packet-drop can be detected just by counting the number of packet-drop. In MANETs, routing misbehavior can severely deteriorate the performance at the routing layer. Specifically, legitimate nodes may involve in the route discovery and maintenance processes but refuse to forward data packets. The acknowledgement-based method [5, 14] will not give enough ground to find the real culprit that is causing packet losses just by counting the number of lost nodes. The fourth method focuses on the different cryptographic methods for improving the detection accuracy. For e.g. the work proposed in [4] provides resource-efficient accountability for node misbehavior. Proofs generated by individual nodes for each forwarding packet is constructed using Bloom filters. Thus, it significantly reduces the communication overhead for misbehavior detection. By analyzing continuous packets one can identify misbehaving nodes based on a series of random audits.

To identify a malicious packet dropping node, the interrelationship between lost packets can be analyzed. More specifically, to ensure the accurate validation of these correlations, the work in [15] proposed a homomorphic linear authenticator (HLA) based public auditing architecture. This allows the identifier to verify the truthfulness of the packet loss information reported by nodes.

Furthermore, some of the observations about routing protocols including AASR, ANODR and AODV help in developing the proposed algorithm. It reveals that the signature generation and accurate detection of malicious nodes in those protocols suffers for transmission delay. The different digital signature schemes are analyzed for the effective construction of our algorithm. We made some assumption during the initial phase such that the source node knows all the public keys of nodes along the path through a dynamic route discovery process and also each individual node along the route is having its own key.

3 System Models and Problem Statement

HLA scheme primarily works on the basis of a client–server scenario. This ensures in assigning the responsibility for providing proof of reception to clients. All the above methods do not perform well under high packet dropping circumstances. Our survey leads to a challenging situation where packet drop rate is essential for comparing the presence of suspicious nodes and link failures.

3.1 Network and Channel Models

Consider a random route R_{SD} in a mobile ad hoc network as shown in Fig. 1. Here we consider our network channel as an infinitely slow speed network with low coverage of mobile nodes. In our system, source node S forwards data to the destination node D through an uninterrupted medium. The intermediate nodes are from N_1 , …, N_K, where N_K is the K^{th} intermediate node. Malicious dropping in high mobility environment is much higher than low mobility environment such that the detection of malicious node is as much easier due to the high adversarial nature of the above medium. In our model, we assume that the network statistics and the channel features are stable for a continuous period of time.

We choose the bitmap result obtained while calculating the interrelationship between the lost/received packets as input to our algorithm. We generate an association database by

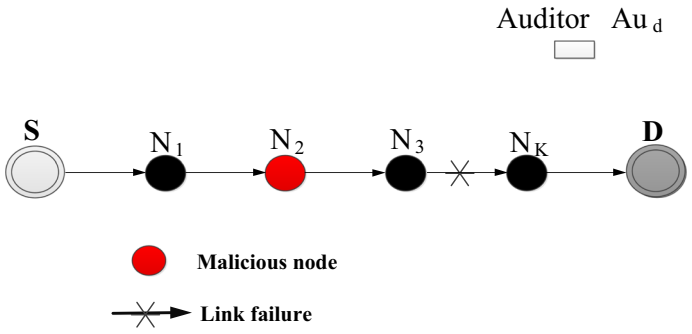

Fig. 1 Network and adversary model

using individual report submitted by nodes with the help of an auditor. The time invariant statistics of nodes are updated on the database by monitoring the status of forwarded packet between intermediate nodes. Our assumption is that a legitimate node always provide truthful information and a malicious node may give false information. The auditor used is unaware of the packet delivered along the route R_{SD} such that we make our auditor independent of the ad hoc network.

3.2 Adversarial Model

The intention of the intruder is to degrade the performance of network by dropping or discarding the packet. Malicious packet dropping can be of any kind such as selective packet drop or a random packet drop. We assume that a malicious node has the full knowledge of routing channel and the detection algorithm for adversarial nodes. We consider the case of multiple malicious nodes along the route R_{SD} including source and destination. So, a malicious node may establish a virtual routing path which is apart from the original routing path and transmits its packet to the downstream malicious node and that kind of data exchange can't be detected by the auditor. So when an auditing is performed, any of the malicious nodes can report a fake reception of packet. This forms an auditing process as vulnerable as possible for accurately predicting whether the packet loss is due to link failure or a malicious drop.

3.3 Problem Statement

Based on the network and adversarial model described above, we can determine the nodes on the routing path that causes the packet drop. We need a coordinator setup which is unaware of the node identities along the route R_{SD}. Privacy conservation during malicious node detection needs to be maintained for packet-reception statistics given by each node. The existing detection mechanism suffers from high communication and storage overhead. In order to avoid the above situation, we need to achieve better detection accuracy and communication overhead. This is obtained by using CLV signature scheme based communal auditing system, so that this can be applicable to a wide variety of wireless ad hoc networks.

4 Proposed Scheme Details

The proposed scheme mainly consists of five phases: network setup phase, data routing phase, communal auditing phase, error node detection phase and data receiver phase.

4.1 Network Setup Phase

The initial setup phase will take place after the route R_{SD} is established. It establishes the path before transmitting any data packets along the route. For establishing the path, the initiator node on the network will send a route request as broadcast towards the destination node. This is done by dynamically with the help of a routing protocol (e.g.; DSR) or by some path finding operation. After receiving the route request from the source, the destination node will reply back to the source which includes the entire route to reach the destination. In this phase, source node, S determines the public keys along the

communication path R_{SD} and the public key is available to all nodes in the path. The transmission of those keys along the route is using the public-key crypto-system based on RSA. Source node S encrypts the message M_1 with the public key of destination node D_j and transmits the cipher text to D_j. Destination node decrypts the cipher text using its secret cipher text list validator (CLV) key.

4.2 Data Routing Phase

Soon after the key generation phase, the source node S enters the data routing phase. S Sends out the CLV signature which includes the message authentication code (MAC) and the actual data for all nodes along the path R_{SD}. During this phase, S encrypts the message using public keys starting from destination node to source node in a reverse manner by computing, $M_i = [(E_D(P_i),E_D(Ch_i)),(E_{NK}(P_i),E_{NK}\ (Ch_i)\ldots.(E_{N1}(P_i),E_{N1}\ (Ch_i))]$,where P_i and Ch_i is the ith data packet of sequence. The checksum value corresponds to each packet will generates the CLV signatures of M_i by padding it with the header of packet. The checksums are generated by using MD5. The header is in the format ($SeqID$, Pkt, $PublicKey$). Let us consider an example for $N = 4$, where 'N' is the no. of packets needed to be send from S to D. As shown in Fig. 1, each node that contains a message block runs CLV signature algorithm using the following steps.

Step 1. For sending a data from S to D, node S splits data into N data packets and determines the combined checksum (CH).
Step 2. The generated checksum is subdivided according to the number of packets.
Step 3. Each packet P_i is appended with a checksum Ch_i, where $i = 1\ldots\ldots\ldots N$
Step 4. The individual packet generated as in Fig. 2 is transmitted to destination D by performing some encryption and decryption processes along the route R_{SD}.

The above encrypted signatures are padded with the message M_i along the route R_{SD} as one-way encrypted chains. By using these hash chains, it intercepts intruders or eaves-droppers from decrypting the message forwarded for the intended receiver. We performed encryption on each resultant packet which is shown in Fig. 3 and the following steps demonstrate the encryption and decryption process done for transferring data from node S to N_1.

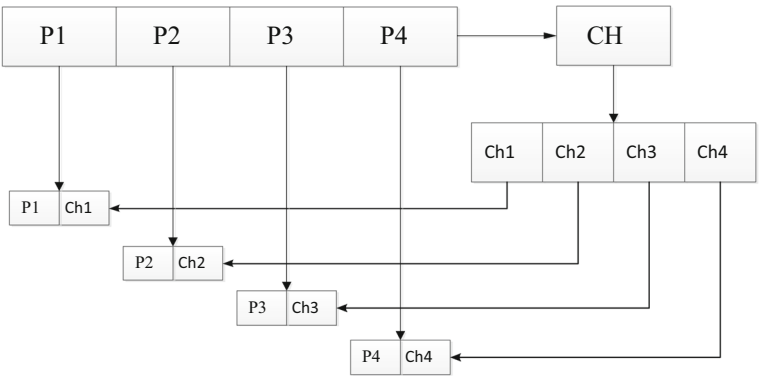

Fig. 2 Packet hashing with checksum values

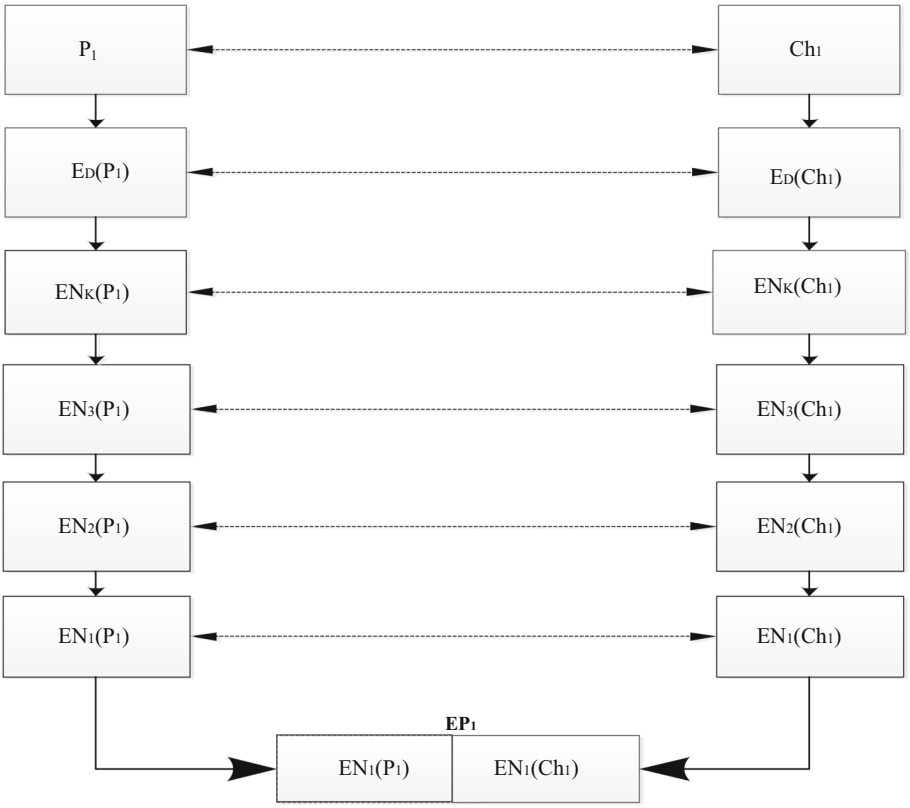

Fig. 3 Encryption at S for node N_1

4.2.1 Encryption at Source 'S'

Step 1. *S* encrypts packet P_1 and the corresponding checksum Ch_1 with the public key of destination *D*.

Step 2. *S* encrypts packet P_1 and the corresponding checksum Ch_1 with the public key of predecessor node of *D* and so on.

Step 3. After completing above steps for all nodes along the route R_{SD}, *S* starts to forward the packet towards node N_1.

4.2.2 Decryption at Node 'N_1'

In our example, node N_1 decrypts encrypted packet EP_1 using its own signature key and forwards it to next node N_2. The authenticity of the retrieved data at the intermediate node is ensured by validating the tag and the CLV signature. The tag value of a message indicates the next node in which the data to be transferred. The above described encryption and decryption is performed repeatedly for all nodes as shown in Fig. 4. It shows that after a series of encryption and decryption process, the actual data to the destination *D* is secure through the above proposed CLV signature scheme.

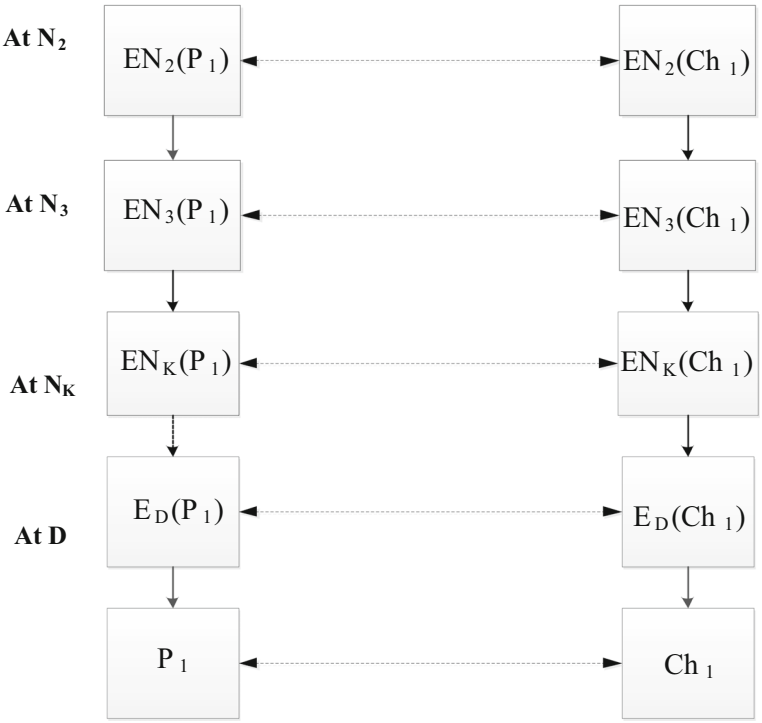

Fig. 4 Packet forwarding from N_1 to D

The above mentioned scheme only describes the transfer of packet P_1 from S to D. Those steps used for transferring first packet are repeated for the remaining packets P_2, P_3, and P_4. Node S computes the received checksum for each step using the hash function f_{Ch1}. Each tag $t_{1i}, \ldots, t_{ji}$ is appended for $j = 1, \ldots, K$ with data $d_1 \ldots \ldots d_n$. The tag t_{ji} is used to verify trustfulness of the received packet along the route.

After getting the packet and the tag value from source S to node N_1, it extracts the actual packet and the checksum. The trustiness of accepted packet is verified by comparing the equality of received checksum with the computed CLV signature, which can be denoted as,

$$Ch_1(M_1) = f_{Ch1}(M1) \tag{1}$$

When the above comparison is a successful one, then the node N_1 can decrypt M_1 in the following manner.

$$D_1(M_1) = M_1 \cap t_{2i} \tag{2}$$

where notation '$\cap$' indicates concatenation operation. After extracting, node N_1 stores the data and the corresponding CLV signature to its routing database called proof-of-acceptance database. This is a dynamic database and is maintained at all the nodes along the path R_{SD}. If the comparison of Eq. (1) fails, N_1 assigns data d_i to the proof-of-acceptance as a lost packet. The above steps are repeated at every intermediate node along the path R_{SD}. Due to the usage of an end-to-end encrypted chain, the last node N_K, only forward the data packet d_i to the destination node D. In general, the flow of CLV signature along different nodes is denoted as follows:

$$\begin{array}{ll} \{E_{N1}(E_{N2}(E_{N3}(E_{NK}(MAC), DATA(D)\} & S \rightarrow N_1 \\ \{E_{N2}(E_{N3}(E_{NK}(MAC), DATA(D)\} & N_1 \rightarrow N_2 \\ \{E_{N3}(E_{NK}(MAC), DATA(D)\} & N_2 \rightarrow N_3 \\ \{E_{NK}(MAC), DATA(D)\} & N_3 \rightarrow N_K \\ \{DATA(D)\} & N_K \rightarrow D \end{array} \quad (3)$$

The above steps show that the security of packet during each transmission is ensured by removing the outer layer of combined data and *MAC*.

4.3 Communal Auditing Phase

This phase is in idle state during the data transmission along the route R_{SD}. The phase is hooked up when they receive a malicious node attack request message from any node along the route. Auditor CA_{ud} receives packet statistics from each node. Each report is send periodically towards the auditor rather than sending notification to source node 'S', why because a malicious node in the reverse path can drop or manipulate the information. The message contains the IDs of nodes on R_{SD}, sequence IDs of recently forwarded packets by S, CLV signature of S and the subset sequence numbers of recent M packets which is received by D. Based on the above information CA_{ud} generates report as shown in Table 1. It shows audit report generated by auditor for nodes N_1 ..... N_4 data transmission. The previous node field in the above table indicates the previous node presented at the time of decryption error. Table shows that node N_2 is the previous node of node N_1 during the occurrence of error. Node count field in table shows the count of occurrence of a node during error.

A random combination of checksum and the data packet is calculated for M packets. The result obtained such as data, signature and bitmap d_{bj} is given to the auditor as an evidence of received packet. Auditor checks the trustiness of data packet and signature by checking whether the value generated with the current bitmap is true for all the packets generated in the old one. If this test is succeeded, then auditor accepts all the packets in d_{bj} and if it fails, auditor states that not all packets generated in the bitmap d_{bj} are actually accepted by N_j. So based on the above result, auditor judge that the node N_j is a harmful node. The above auditing scheme not only guarantees that a node cannot downplay its packet loss but also they cannot state the reception of a message which is not actually received by it. This is intercepted by using the communal auditor CA_{ud}.

If previous count of a node is greater than zero, then it is a false node, otherwise check the false probability F_p by the following eqn.

$$F_p = \frac{Fail}{Total\ no\ of\ packets\ forwarded}$$

If F_p is greater than above a threshold, then there is a chance of link failure.

Table 1 Auditing information of participating nodes along the route

Node	Participating nodes	Success	Fail	Error	Previous node	Node count
N_1	10	7	3	1	N_2	0
N_2	25	16	9	0		1
N_3	40	26	14	1	N_1	0
N_4	35	22	13	0		1

4.4 Error Node Detection Phase

During this phase, the error nodes are detected based on the correlation between messages that combines the different bitmap values for the data reception. The CA_{ud} enters the detection phase after receiving and auditing the reply to its query from all nodes on R_{SD}. The main functions of CA_{ud} in this phase includes: detecting any overstatement of packet loss at each node, generate a packet-loss bitmap for each node, calculating the autocorrelation function for the packet loss on each node, and decides whether malicious behavior is present or not. More specifically, CA_{ud} performs these tasks as follows.

The auditor calculates the autocorrelation function. The detection process applies on an end-to-end path. The detection for diverse paths can be performed as different independent detections for each path. Although the optimal error threshold that reduces the detection error is still an open problem. Our simulations show that through trial-and-error, one can easily find a good threshold εth that provides better detection accuracy than the optimal detection scheme that utilizes only the pad of the number of lost packets. In a data-reception bitmap, lost packet and received packet is denoted by the bit value '0' and '1' respectively. More specifically, consider the above Table 2.

4.5 Data Receiver Phase

All participatory nodes send individual reports of received and forwarded information to the communal auditor. Here in our framework, node *D* sends an acknowledgement towards the auditor. The received acknowledgement is verified as explained in Sect. 4.3 and submits back to the destination.

5 Security Analysis

By using some lemma that helps to prove our proposed scheme is a valid one. On the basis of these lemmas, we proved that our scheme provides better security by classifying the reason for different attacks during the detection of malicious nodes. Following are the different conditions that can be used for analyzing the security of our scheme.

Lemma 1 *If we cannot decrypt the received data from a node, then that indicates the presence of a malicious node.*

Lemma 2 *If a packet is dropped during the data transmission between S and D, then we state that there occurs both the presence of malicious node as well as a link failure.*

Lemma 3 *If a packet is received and the checksum verification (received checksum, Ch_1 ==calculated checksum, f_{Ch1}) is failed, then our scheme defines that there is a malicious node is located along the route R_{SD}.*

Table 2 Packet-reception bitmap for malicious node detection

	N_1	N_2	N_3	N_4	Malicious node	Status
P_1	1	0	1	1		Success
P_2	1	1	1	0	N_2	Fail
P_3	1	1	0	1	N_4	Fail
P_4	0	1	1	1		Success

6 Overhead Analysis

The scheme which is proposed requires high computation cost due to the encryption process takes place at the source node, but it achieves high detection accuracy of a malicious node and low storage overhead along the route and helps in improving the trusted detection accuracy of malicious nodes during communication.

6.1 Computation Overhead

Our CLV signature scheme follows modified El Gamal signature scheme which is useful in discrete logarithmic problems. By using El Gamal encryption, we are generated CLV signatures for all nodes along the route R_{SD} for each packet. This causes CLV signature generation process as a tedious one. Thus the computation overhead at source node S becomes vulnerable. This problem can be avoided by making our signature scheme scalable as the network size increases. Since the communal auditor is independent of the ad hoc medium and it preserves privacy without revealing the node information. We assume the communal auditor as the dedicated service provider and thus the computation overhead for auditing process will not affect the nodes communicating on network.

6.2 Communication and Storage Overhead

The communication overhead occurs during the reception and forwarding at individual nodes along the route R_{SD}. The public key generation and checksum generation at source node incurs a one-time cost and they acquire the information with the help of a dynamic source routing protocol. The checksum generated for each individual packet is using random combination of 128-bit long MD5 packet. Each node along the path generates individual reports based on packet reception and forwarding and submits report to the communal auditor,CA_{ud}. The reports are generated periodically as beacon messages rather than sending it on sequentially and thus it reduces the storage overhead of auditing process. Here the length of encrypted packet is high at the source and it becomes low when it reaches towards destination. Thus, it indicates that the computation and communication cost at source node is high while compared to intermediate nodes.

7 Performance Evaluation

7.1 Simulation Setup and Results

The detection accuracy of our proposed scheme can be achieved by the Conventional algorithm with the optimal maximum likelihood algorithm that utilizes the distribution of number of lost packets. We create a simulation environment using one simulator (version one_1.4.1).We setup a network consist of 150–200 nodes. Out of these 200 nodes 10 % is set as malicious and generate 5 % of link failures during packet transmission. We applied the proposed detection algorithm under various network sizes and various percentages of defective nodes. Based on the information we simulated the system and the results are plotted in Figs. 5 and 6.

Suppose we are sending 150 packets.Out of these 100 packets were successful ones and 50 packets were failure ones. We monitored the continuous packet drops for subsequent

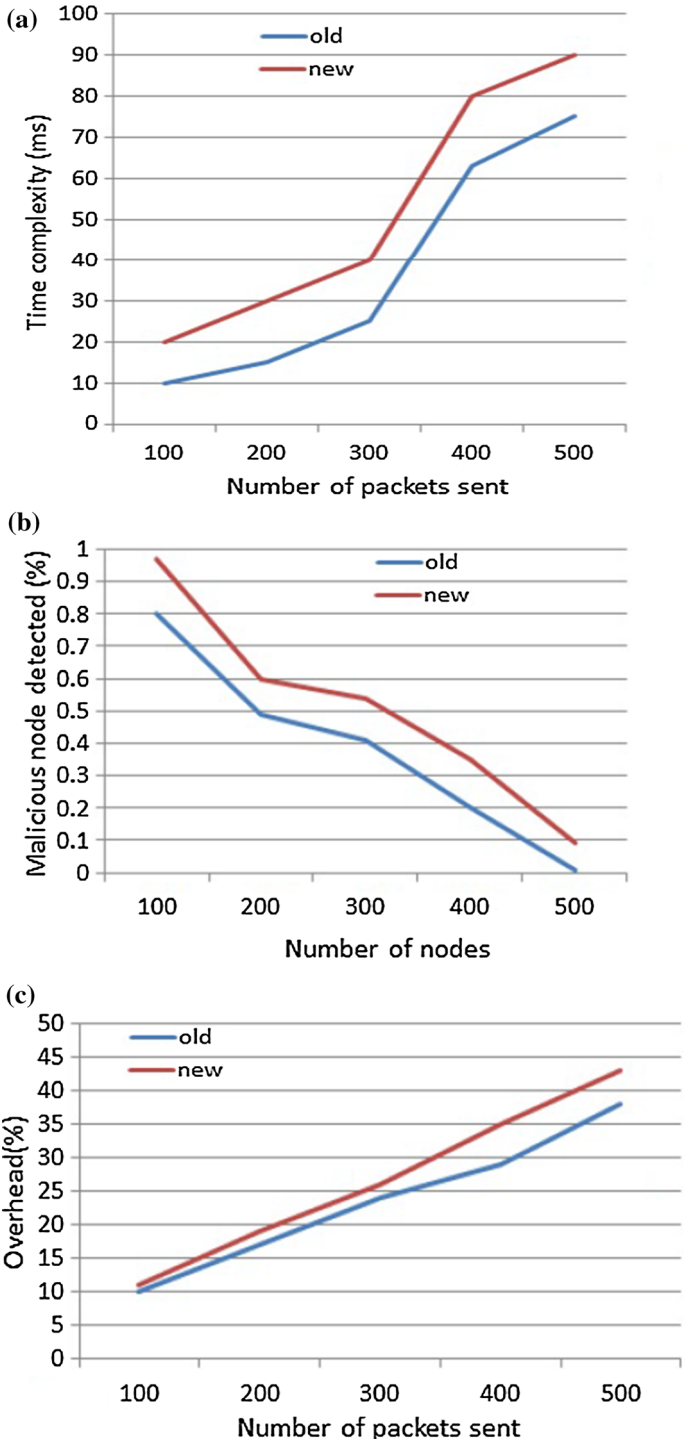

Fig. 5 Performance comparison under different network sizes

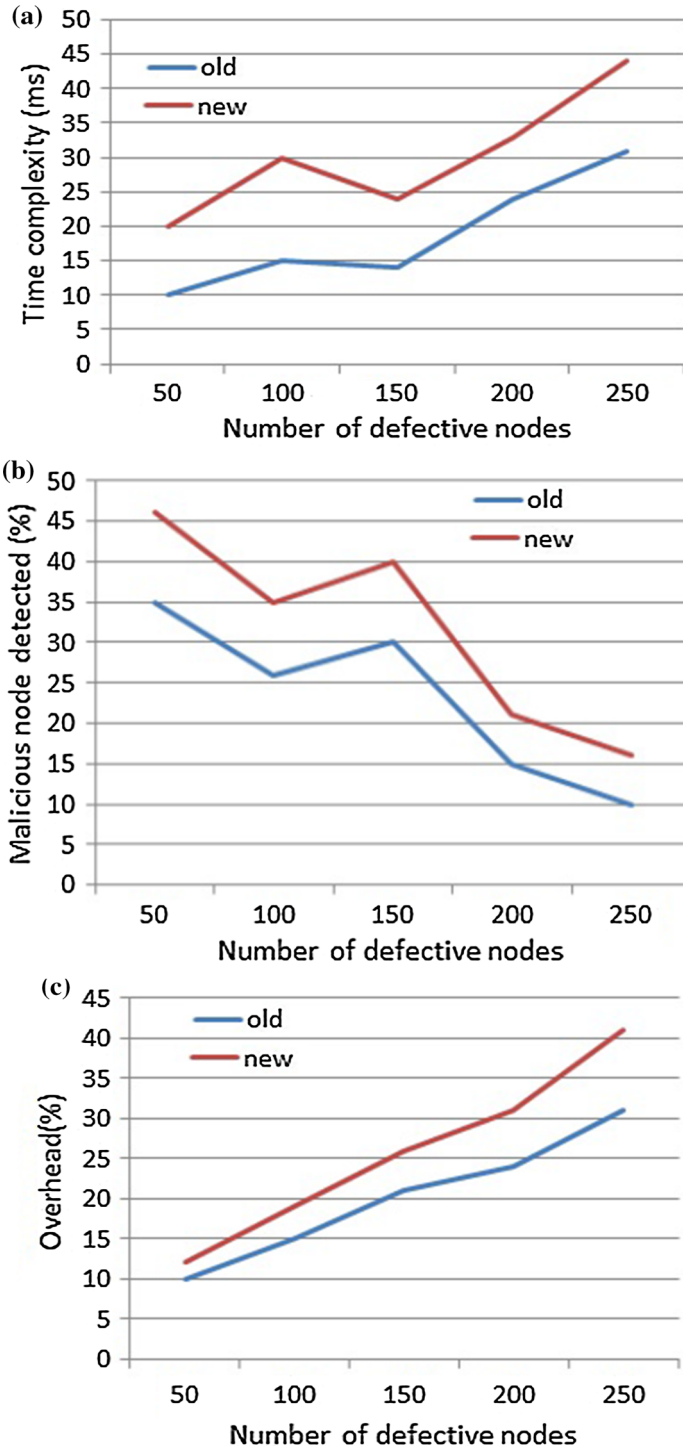

Fig. 6 Performance comparison under different percentage of defective nodes

nodes and analyze the results based on successful and unsuccessful reception of packets. According to the variation under different network sizes including 100,200 and 300 nodes are verified. While observing Fig. 5a, we show that when the size of packet increases, time taken for data transmission will also be increased due to the high signature generation. The detection accuracy and detected rate of proposed scheme is highly improved which is shown in Fig. 5b. The storage overhead is low while compared to the old scheme, but the communication overhead is slightly higher than old one. Figure 5c shows that when we are sending 100 packets, the overhead percentage is 14, 15 for old scheme and new scheme respectively. Overhead is always slightly higher over increase in number of sending packets.

The Fig. 6 shows the performance comparison in the presence of different percentage of defective nodes for various parameters such as time complexity,overhead and malicious node detection rate. The time complexity is high while increase in number of defective nodes during packet transmission. Figure 6a shows that time complexity is high due to the end-to-end delay in data transmission. Figure 6b depicts the malicious node detection rate against the number of defective nodes. Presence of malicious nodes will not give much effect in the variation of overhead as shown in Fig. 6c.

8 Conclusion and Future Work

In this paper, we present a comparative analysis of the proposed scheme with other detection algorithms and it improves the accuracy of detecting malicious nodes by utilizing the interrelationship between dropped packets. The new scheme accurately calculates the packet drop due to malicious node and link failure separately, which is significant to accept trustful packet-drop information at each node. The CLV signature based communal auditing framework is helpful in ensuring the integrity of packet-drops informed by individual nodes. This is because the comparison between the number of dropped packets in the case of link-failure-only case and the link-failure-plus-malicious-dropping case seems to be feeble. This framework achieves low communication and storage overhead along intermediate nodes while compared to the previous work. It uses summary of individual report after receiving all packets and sends it to the auditor periodically as beacon messages. This will reduce the communication delay for auditing. Detection accuracy of the proposed algorithm increases highly with discriminating malicious drops. To improve the retransmission overhead of the framework, the integration of different signature generation schemes are focused in our future studies.

References

1. Liu, W., & Yu, M. (2014). AASR: Authenticated anonymous secure routing for MANETs in adversarial environments. *IEEE Transaction On Vehicular Technology, 63*(9), 4585–4593.
2. Shu, T., & Krunz, M. (2015). Privacy-preserving and truthful detection of packet dropping attacks in wireless ad hoc networks. *IEEE Transactions on Mobile Computing, 14*(4), 813–828.
3. Zhong, S., Chen, J., and Yang, Y.R. (2003). Sprite: A simple cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of IEEE INFOCOM Conference*, pp. 1987–1997.
4. Kozma Jr. W., & Lazos, L. (2009). REAct: Resource-efficient accountability for node misbehavior in ad hoc networks based on random audits. In *Proceedings of ACM conference wireless network security*, pp. 103–110.

5. Liu, K., Deng, J., Varshney, P., & Balakrishnan, K. (2006). An acknowledgement-based approach for the detection of routing misbehavior in MANETs. *IEEE Trans. Mobile Comput., 6*(5), 536–550.
6. Liu, Y., & Yang, Y. R. (2003). Reputation propagation and agreement in mobile ad-hoc networks. In *Proceedings of IEEE WCNC conference*, pp. 1510–1515.
7. Zhang, Y., Lazos, L., & Kozma, W. (2012). AMD: Audit-based misbehavior detection in wireless ad hoc networks. *IEEE Transactions Mobile Computing, 99*, 1.
8. Reed, M. G., Syverson, P. F., & Goldschlag, D. M. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications, 16*(4), 482–494.
9. Boneh, D., Boyen, X., & Shacham, H. (2004). Short group signatures. In *Proceedings of CRYPTO*, pp. 41–55.
10. Yu, M., Zhou, M. C., & Su, W. (2009). A secure routing protocol against Byzantine attacks for MANETs in adversarial environment. *IEEE Transactions on Vehicular Technology, 58*(1), 449–460.
11. Awerbuch, B., Holmer, D. Rotaru, C.-N., & Rubens, H. (2002). An on-demand secure routing protocol resilient to byzantine failures. In *WiSE '02 proceedings of the 1st ACM workshop on wireless security*.
12. Buchegger, S., & Boudec, J.-Y. L. (2002). Performance analysis of the CONFIDANT protocol: Cooperation of nodes—fairness in dynamic ad-hoc networks. In *Proceedings of IEEE/ACM workshop on mobile ad hoc networking and computing (MobiHOC)* (Online). http://lcawww.epfl.ch/Publications/LeBoudec/BucheggerL02.pdf.
13. Marti, S., Giuli, T., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the sixth international conference on mobile computing and networking 2000*, Boston, MA, August 2000 (Online). http://gunpowder.stanford.edu/laik/projects/adhoc/mitigating.pdf.
14. Balakrishnan, K., Deng, J., & Varshney, P. K. (2005). TWOACK: Preventing selfishness in mobile ad hoc networks. In *Proceedings of IEEE wireless communications and networking conference (WCNC'05)*.
15. Shu, T., & Krunz, M. (2012). Detection of malicious packet dropping in wireless ad hoc networks based on privacy-preserving public auditing. In *Proceedings of WiSec*, pp. 87–98.
16. William, S., & Stallings W. (2009). *Cryptography and network security* (4th ed.) Delhi, India: Pearson Education India, Eighth Impression.
17. Chou, C.-C., Wei, D. S. L. Jay, Kuo, C.-C., & Naik, K. (2007). An efficient anonymous communication protocol for peer-to-peer applications over mobile ad-hoc networks. *IEEE Journal on Selected Areas in Communications, 25*(1), 192–203.
18. Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad hoc on-demand distance vector (AODV) routing. In *IETF RFC 3561* (online). www.ietf.org/rfc/rfc3561.txt.
19. Kong, J., Hong, X., & Gerla, M. (2007). ANODR: An identity-free and ondemand routing scheme against anonymity threats in mobile ad hoc networks. *IEEE Transactions on Mobile Computing, 6*(8), 888–902.
20. Johnson, D., Hu, Y., & Maltz, D. (2007).The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. In *IETFRFC4728* (online). www.ietf.org/rfc/rfc4728.txt.
21. Shen, H., & Zhao, L. (2013). ALERT: An anonymous location-based efficient routing protocol in MANETs. *IEEE Transactions on Mobile Computing, 12*(10), 1079–1093.
22. Shu, T., Krunz, M., & Liu, S. (2010). Secure data collection in wireless sensor networks using randomized dispersive routes. *IEEE Transactions on Mobile Computing, 9*(7), 941–954.
23. Proano, A., & Lazos, L. (2012). Packet-hiding methods for preventing selective jamming attacks. *IEEE Transactions on Dependable and Secure Computing, 9*(1), 101–114.

M. S. Rahul is a PG (M.Tech) Scholar in Computer Science and Engineering, TKM Institute of Technology, affiliated to Cochin University of Science and Technology. His research area includes security in Next Generation Networks and MANET.

E. Arun received Ph.D. degree in Information and Communication Engineering from Anna University, Chennai. Currently he is senior Professor in the Department of Computer Science and Engineering at TKM Institute of Technology, Kerela. He teaches Post Graduate level courses on wireless communication systems. He has completed two grant in-aid research projects in the field of Mobile Communications. His research interest includes Cross Layer Mobility Management, Cognitive Radio and Cloud Computing.

P. Mohamed Shameem received Ph.D. in Computer Science and Engineering from Noorul Islam University, Tamil Nadu. Currently he is working as Professor in Department of Computer Science and

Engineering at TKM Institute of Technology, Kerela. His area of research includes Cloud Computing and Next Generation Networks.

J. Rajeesh received Ph.D. degree in Information and Communication Engineering from Anna University, Chennai. He is working as Senior Professor in the Department of Electronics and Bio Medical Engineering at TKM Institute of Technology, Kerela. His area of research includes Medical Imaging and Next Generation Networks.